What is claimed is:

1   1.   A process for use in a database system, comprising:

2   storing data according to a first user-defined data type in a table;

3   associating at least a first compression routine with the first user-defined data

4   type; and

5   using the first compression routine to compress the data according to the first

6   user-defined data type.

1   2.   The process of claim 1, further comprising using a second compression routine to

2   compress the data to improve compression efficiency.

1   3.   The process of claim 2, wherein using the first and second compression routines

2   comprises using user-defined data type methods.

1   4.   The process of claim 3, wherein using the user-defined data type methods

2   comprises using methods built in with the first user-defined data type.

1   5.   The process of claim 1, wherein using the first compression routine comprises

2   using a first compression method built in with the first user-defined data type.

1   6.   The process of claim 5, further comprising providing a user-defined method

2   executable to invoke the first compression method.

1   7.   The process of claim 6, further comprising invoking the user-defined method to

2   invoke a second compression method built in with the first user-defined data type.

1   8.   The process of claim 7, wherein invoking the user-defined method comprises

2   invoking the user-defined method to alter compression efficiency.

11

1   9.   The process of claim 1, further comprising providing a second user-defined data

2   type built upon the first user-defined data type.

1   10.   The process of claim 9, further comprising storing a first type of data using the

2   first user-defined data type and storing a second type of data using the second user-

3   defined data type.

1   11.   The process of claim 10, further comprising using a second compression routine

2   to compress the second type of data.

1   12.   The process of claim 9, further comprising inheriting at least a data structure and

2   at least a built-in method from the first user-defined data type into the second user-

3   defined data type.

1   13.   An article comprising at least one storage medium containing instructions that

2   when executed cause a system to:

3   store data according to a first user-defined data type; and

4   associate a first compression routine with the first user-defined data type for

5   compressing the data.

1   14.   The article of claim 13, wherein the instructions when executed cause the system

2   to associate a second compression routine with the first user-defined data type, the first

3   and second compression routines providing different compression algorithms.

1   15.   The article of claim 14, wherein the instructions when executed cause the system

2   to provide the first compression routine as a method built in with the first user-defined

3   data type.

1   16.   The article of claim 15, wherein the instructions when executed cause the system

2   to provide the second compression routine as a method built in with the first user-defined

3   data type.

1   17.   The article of claim 13, wherein the instructions when executed cause the system

2   to associated a first data structure with the first user-defined data type, the first data

3   structure to indicate a type of compression applied on a data object.

1   18.   The article of claim 17, wherein the instructions when executed cause the system

2   to associate a second data structure with the first user-defined data type, the second data

3   structure to indicate a percentage amount of compression of the data object.

1   19.   The article of claim 18, wherein the instructions when executed cause the system

2   to access the first and second data structures of the data object when accessing the data

3   object.

1   20.   The article of claim 19, wherein the instructions when executed cause the system

2   to store the data object in a relational table.

1   21.   The article of claim 19, wherein the instructions when executed cause the system

2   to store the data object in a relational table distributed across multiple access modules.

1   22.   The article of claim 20, wherein the instructions when executed cause the system

2   to provide a second user-defined data type built upon the first user-defined data type.

1   23.   The article of claim 13, wherein the instructions when executed cause the system

2   to provide a second user-defined data type built upon the first user-defined data type.

1   24.   The article of claim 23, wherein the instructions when executed cause the system

2   to inherit the first compression routine from the first user-defined data type into the

3   second user-defined data type.

1 25. The article of claim 24, wherein the instructions when executed cause the system

2 to:

3   associate a second compression routine with the first user-defined data type; and

4   inherit the second compression routine from the first user-defined data type into

5 the second user-defined data type.

1 26. The article of claim 25, wherein the instructions when executed cause the system

2 to:

3   store a first type of data using the first user-defined data type; and

4   store a second type of data using the second user-defined data type.

1 27. A database system, comprising:

2   a storage system to store at least a table;

3   a plurality of compression routines to apply respective different compression

4 algorithms; and

5   a controller adapted to invoke one of plurality of compression routines to

6 compress data stored in the table.

1 28. The database system of claim 27, wherein the table includes a relational table and

2 the data is stored in a first attribute of the relational table.

1 29. The database system of claim 28, wherein the first attribute is according to a first

2 user-defined data type.

1 30. The database system of claim 29, wherein the plurality of compression routines

2 are methods built in with the first user-defined data type.

1 31. The database system of claim 30, the storage system to store a second table

2 having a second attribute according to a second user-defined data type built upon the first

3 user-defined data type.

1   32.   The database system of claim 27, wherein the controller is adapted to invoke

2   another one of the compression routines to alter compression of the data.

1   33.   The database system of claim 32, wherein the controller is adapted to invoke

2   another one of the compression routines in response to a Structured Query Language

3   UPDATE statement.

1   34.   The database system of claim 33, wherein the controller comprises a user-defined

2   method.

1   35.   The database system of claim 34, wherein the plurality of compression routines

2   comprise methods built in with the first user-defined data type,

3   the user-defined method executable to invoke the methods built in with the first

4   user-defined data type.

1   36.   The database system of claim 27, further comprising a plurality of access modules

2   adapted to manage access to respective portions of the storage system.

1   37.   The database system of claim 36, wherein the table is distributed across multiple

2   access modules.